Dynamic Channel Assignment in Cellular Radio

Kumar N. Sivarajan and Robert J. McEliece*
116-81, California Institute of Technology
Pasadena, CA 91125
and
John W. Ketchum
GTE Laboratories Inc.
Waltham, MA 02154

Abstract

This paper describes some dynamic channel assignment algorithms for cellular systems, that we have developed. These algorithms are shown to be feasible for implementation in current cellular systems while their performance, in some cases, is close to the best achievable by any channel assignment algorithm. For the examples considered, in the interesting range of blocking probabilities (2 – 4 %), the dynamic channel assignment algorithms have yielded an increase of 60 – 80 % in the carried traffic, over the best known fixed channel assignment.

Introduction

In fixed channel assignment (FCA), only a fixed fraction of all the channels is available in each cell whereas in dynamic channel assignment (DCA), *all* the channels are available in *all* the cells. The objective in DCA is to develop a channel assignment strategy which minimizes the total number of blocked calls and is, in addition, feasible for implementation. We present several DCA algorithms and compare their performance with an easily simulated bound that is similar to the 'maximum packing' bound mentioned in [1]. Using this bound, we demonstrate that in the case of homogeneous spatial traffic distribution, some of these algorithms are virtually unbeatable by *any* channel assignment algorithm. Another important feature of these algorithms is that, unlike some of the previously reported DCA algorithms [1], they are better than FCA even for heavy traffic.

* The work of K. N. Sivarajan and R. J. McEliece was supported by a grant from GTE Laboratories Inc. The work of R. J. McEliece was also partially supported by a grant from Pacific Bell.

Notation :

N : the number of cells in the system;

c_{ij}, $1 \leq i,j \leq N$: the frequency separation required between a call in cell i and a call in cell j;

n_i, $1 \leq i \leq N$: the number of calls in progress in cell i;

p_i, $1 \leq i \leq N$: the probability that a new call arrives in cell i;

ρ : the total traffic in the system;

$\rho_i = p_i\rho$, $1 \leq i \leq N$: the traffic in cell i;

N_f : the number of (contiguous) frequency channels available. These channels are numbered 1 through N_f.

f_{ik}, $1 \leq i \leq N$, $1 \leq k \leq m_i$: the frequency assigned to the kth call in the ith cell.

Some more notation will be introduced later, as the need arises.

Constraints :

$|f_{ik} - f_{jl}| \geq c_{ij}$ for all i, j, k, l except for $i = j$, $k = l$.

Assumptions :

- Call arrivals in cell i are independent of all other arrivals and obey a Poisson distribution with parameter ρ_i.
- Call holding times are exponentially distributed with mean call duration = 180 secs.
- There are no calls handed off between cells.
- Blocked calls are cleared (ErlangB).

The assumption of memoryless arrivals and holding times is standard in wireline telephony and it is reasonable to assume that these hold for mobile telephony as well.

Four dynamic channel assignment strategies will be considered. They are :

1. *Simple* : An incoming call is assigned the least available frequency.
2. *Maxavail* : Of all the frequencies that can be assigned to an incoming call, the frequency which

maximizes the total number of channels available in the entire system, is assigned to an incoming call.

3. *Remax1* : If no frequency is available for assignment to an incoming call (using the Maxavail strategy), one of the calls in progress is permitted to be reassigned to a different frequency. The reassignment is also carried out using the 'Maxavail' principle.

4. *Remax2* : If no channel is available for assignment to an incoming call using the 'Remax1' strategy, one more call is permitted to be reassigned to a different frequency.

The details of the simulation and the channel assignment algorithms will now be described in some detail. Since the arrival process is assumed to be memoryless, the time between call arrivals is exponentially distributed. The Mean Time Between call Arrivals (MTBA) in the entire system equals $(180/\rho)$ seconds, since the mean call duration is assumed to be 180 seconds. To simulate the process of call arrivals and departures, the following steps are carried out :

Time is discretized to steps of 10 ms. Minimum duration of a call, and the minimum time between call arrivals, are assumed to be one time step. This makes the simulation slightly easier.

1. To start the process generate an exponentially distributed random variable (ERV) with mean = MTBA.
2. Increment the time by one step and check for a call arrival. If there are no arrivals, go to step 9.
3. Choose a cell for the call with the required distribution (i.e., choose cell i with probability p_i).
4. Generate an ERV with mean = MTBA for the time to the next call arrival.
5. Call the channel assignment routine.
6. If it returns a channel, assign it to the call, and also make the required call rearrangements.
8. Generate an ERV with mean 180 seconds for the call duration, and keep track of the channels that are not assignable to future calls in each cell because of this assignment.
9. Check for call departures.
10. If there are any departures, free all the channels that were tied up in each cell because of the frequencies assigned to the departing calls.
11. Go to step 2.

Each of the following channel assignment algorithms requires that the cell in which the call arises, referred to as 'callcell' below, be passed to them. They can access all the existing assignments (say, through global variables). They either return a channel that can be assigned to the call, or report, 'call blocked'. If it is a rearrangement strategy, the required call rearrangements are also returned.

Simple :

1. From the list of channels 1 through N_f in 'callcell', return the first channel that is free .
2. If there is no free channel, report 'call blocked'.

Maxavail :

1. For each available channel in 'callcell', compute

Systemwide Channel Availability

$$= \sum_{\text{all cells}} \text{Number of available channels} \quad ,$$

assuming that this channel is assigned to the call.
2. Return that channel which maximizes this sum.
3. In the case of a tie, return the least channel.
4. If no channel is available in 'callcell', report 'call blocked'.

Remax1 :

1. Call 'Maxavail'.
2. If 'Maxavail' reports 'call blocked', go to step 3. Otherwise, return the channel returned by 'Maxavail'.
3. Make a list of all the channels in 'callcell' that are unavailable because of exactly one other call that is in progress.
4. Call 'Maxavail' for each of these calls (interferers).
5. For each of the interferers that are not reported 'blocked' by 'Maxavail', by assuming that the channel returned by 'Maxavail' is assigned to them, and that the corresponding 'freed' channel is assigned to the new call, compute 'Systemwide Channel Availability'
6. Return that interferer which maximizes this sum, the channel to which it should be reassigned, and the channel that is to be assigned to the new call.
7. If no channel can be freed by reassigning a single call, report 'call blocked'.

Remax2 :

1. Call 'Remax1'.
2. If 'Remax1' reports 'call blocked', go to step 3. Otherwise, return the channel returned by 'Remax1'.
3. Make a list of all the channels in 'callcell' that are unavailable because of exactly one other call that is in progress.
4. Call 'Remax1' for each of these calls (interferers).
5. For each of the 'interferers' that are not reported 'blocked' by 'Remax1', by assuming that the rear-

rangement required by 'Remax1' is made, the channel returned by 'Remax1' is assigned to the 'interferer', and that the corresponding 'freed' channel is assigned to the new call, compute 'Systemwide Channel Availability'

6. Return the two rearrangements, and the freed channel, corresponding to that call which maximizes this sum

7. If no channel can be freed using this strategy, report 'call blocked'.

The performance of these channel assignment strategies is now investigated. The cellular system that is chosen for this purpose is the $N = 21$ system described in [2], p.13. The channel assignment constraints are : co-channel constraints equivalent to a 12-cell cluster, adjacent channel constraints for adjacent cells and a co-site constraint of 5. The cellular system is reproduced in Fig. 1.

Case 1 : Homogeneous Spatial Traffic Distribution

In this case, $p_i = (1/N)$ for all i. The number of channels available is $N_f = 96$. Because of the constraints chosen, this implies that one can only assign 8 channels per cell, under a fixed channel assignment scheme. The operation of the cellular system is simulated for a period of 3 hours and the average blocking, which is the ratio of the number of blocked calls and the number of call attempts, is plotted as a function of the increase in traffic, in Fig. 2.

It is readily seen that the strategies can be ranked in decreasing order of performance as follows : Remax2, Remax1, Maxavail and Simple. In principle, of course, the best strategy is one that permits everyone to be rearranged to accommodate a call. This is equivalent to solving the fixed channel assignment problem every time an incoming call is blocked. This would not only involve a great deal of computation – even if the fast heuristic algorithms developed in [4] are used – but also result in a considerable decrease in service quality, as there would be a large number of channel reassignments per incoming call. In contrast, Remax2 reassigns *at most* 2 calls to accommodate a new call, and only after attempting to accommodate it without reassignments, or with one reassignment only. The natural question to be asked is :" What is the difference in performance between 'Remax2' and this 'optimal' strategy?"

The evaluation of the optimal strategy involves a lot of computation. Hence, a lower bound on the performance of this strategy is obtained with the help of lower bounds on fixed channel assignments described in [2]. This is plotted as 'Bound' in Fig. 2. In the computation of this bound, an incoming call is blocked if, and only if, accommodating this call would mean that, the lower bound, on the number of channels required to handle this configuration of calls, exceeds $N_f = 96$.

We were pleasantly surprised to find that, in this case, the performance of 'Remax2', in terms of minimizing the average blocking, is close to the *best that can be achieved using any conceivable strategy.*

Case 2 : Inhomogeneous Spatial Traffic Distribution

The p_i in this case are obtained by treating the channel requirements specified in [2] for the example under consideration, as relative traffic densities. These p_i are listed in Table 1. The performance of these algorithms is plotted in Fig. 3. Once again, of the channel assignment strategies considered, Remax2 is the best; but Remax1 is almost as good. But, unlike Case 1, the performance of 'Remax2' is not extremely close to 'Bound'. This difference in performance could either be because 'Remax2' is not close to the best that one can do in this case, or because 'Bound', which is only a *lower bound* on the best performance achievable, is not tight enough. In the Appendix, it is demonstrated, with the help of a simple example, that this lower bound tends to become less tight as the inhomogeneity in the traffic increases. This is encouraging; it suggests that 'Remax2' may actually be near-optimal in this case, too.

It is also worth noting that, for the same traffic, the average blocking in this inhomogeneous system is considerably more than that in the homogeneous system. This seems to confirm the widely held belief among cellular system designers that, the traffic in the system should be as homogeneous as possible. ("... matching the spatial density of available channels to the spatial density of demand for channels ...", V.H.MacDonald in [3], p. 19).

Next, the question of computing a lower bound on system performance, *without simulation*, will be addressed.

The process of call arrivals and departures is a discrete-state, continuous-time, Markov process, because of the assumption of memoryless arrivals and holding times. The state of the system is specified by the N-dimensional vector $\tilde{n} = (n_i)$ whose components are the number of calls in progress in the N cells in the system. In particular, the Markov process is a birth-death process since only transitions to neighbouring states (in N-dimensional space) are permitted, i.e., the probability of multiple arrivals, or departures, or an arrival and a departure, in a small time interval Δt, is negligible, compared to the probability of a single arrival or departure. The allowed states of the system are all the states $\tilde{n} = (n_i)$, which can possibly exist, with the given number of channels and the constraints on their assignments. Let $\mathcal{N}$ denote the set of all allowed states of the system. From these assumptions it can be shown that the equilibrium probability that the system is in state $\tilde{n}' = (n_i'), 1 \leq i \leq N$, is given by

$$P_{\tilde{n}'} = \frac{\rho_1^{n_1'} \rho_2^{n_2'} \ldots \rho_N^{n_N'} / n_1'! n_2'! \ldots n_N'!}{\sum_{\mathcal{N}} \rho_1^{n_1} \rho_2^{n_2} \ldots \rho_N^{n_N} / n_1! n_2! \ldots n_N!} .$$

For the computation of the lower bound on the blocking probability, it has already been assumed that the set of allowed states is specified by inequalities of the form

$$\sum_{j=1}^{k} n_{i_j} \leq N_{\text{clique}}$$

where the cells i_j, $j = 1, 2, \ldots, k$, form a clique in which the maximum number of calls that can be in progress is N_{clique}.

A *ν-clique* is a maximal set of cells, in which, any two frequencies used should be separated by at least ν. Formally, if X denotes the set of all cells in the system, $Q \subseteq X$ is called *ν-complete* if $c_{ij} \geq \nu \quad \forall i, j \in Q$. (Refer to [2]). A *ν-clique* is a *maximal*, *ν-complete* subset of X. In the system under consideration, there are 1-cliques, 2-cliques and 5-cliques. Each cell by itself constitutes a 5-clique because of the co-site constraint of 5, and any set of three mutually adjacent cells constitutes a 2-clique, because of the adjacent channel constraint on adjacent cells. The 1-cliques in the system are listed in Table 2. For $N_f = 96$, the maximum number of calls that can be in progress in a single cell is 20, in a 2-clique is 48 and in a 1-clique is 96.

If the set of cells $Cl = \{i_1, i_2, \ldots, i_k\}$ constitutes a clique in which the maximum number of calls that can be progress is N_{clique}, the probability that this clique is 'saturated', i.e., has N_{clique} calls in progress is given by

$$P_{sat} = \frac{\sum_{\hat{\mathcal{N}}} \rho_1^{n_{i_1}} \rho_2^{n_{i_2}} \ldots \rho_N^{n_{i_N}} / n_{1_1}! n_{i_2}! \ldots n_{i_N}!}{\sum_{\mathcal{N}} \rho_1^{n_1} \rho_2^{n_2} \ldots \rho_N^{n_N} / n_1! n_2! \ldots n_N!} ,$$

where $\hat{\mathcal{N}} = \{\mathcal{N} \mid n_{i_1} + \ldots + n_{i_k} = N_{\text{clique}}\}$.

This quantity is difficult to compute if there are more than two overlapping cliques. In order to compute this quantity, it is assumed that there are only *two* cliques in the system, i.e., the other cliques are permitted to have any number of calls in progress. These two cliques are Cliques 2 and 3 listed in Table 2. The probability that each of these cliques is saturated can be computed. The probability that both of them are saturated can also be computed. These results enable us to compute a lower bound on 'Bound'. This is plotted as 'Calbound' in Figs. 2 and 3.

For the inhomogeneous case, the performance of 'Calbound' leaves much to be desired. It is possible to improve the performance of 'Calbound' by choosing a different set of two cliques — e.g. the 2-clique with the densest traffic consisting of cells 8, 9 and 16 and one of the 1-cliques in which it is contained — but it does not seem likely that a simple (easily calculated), but tight, bound on the blocking probability of a cellular system can be obtained, in the general case, along these lines. Therefore, a different line of approach has to be devised. The authors have only had partial success in this respect so far and pose this as one of the important open problems in this area.

An aspect of the algorithms for channel assignment, that needs to be studied, is their behaviour in the case of time-varying traffic. In the case of extreme spatial non-uniformity, i.e., when all the traffic is in a single cell, the best that can be done is to make the maximum number of channels, viz. 20 in the example considered, available in that cell. All the algorithms described - from Simple to Remax2 - do just that. Therefore, the algorithms can be said to be *robust* w.r.t. spatial non-uniformity. Their robustness w.r.t. temporal non-uniformity has to be investigated, and new algorithms have to be developed, if found necessary.

Then, there is the question of computation. For the example considered, the running time for 'Simple' is about 10 minutes, while 'Maxavail', 'Remax1' and 'Remax2' take between 20 and 25 minutes, on a MIPS M/120 RISComputer. Considering that the performance of these algorithms is simulated for a period of 3 hours of simulated time, these algorithms can be said to run in 'real' time. A cellular system, consisting of 21 cells, is not atypical of existing systems. The number of channels available is typically about 3 to 4 times what had been assumed above. But, the running time of these algorithms is only expected to increase quadratically with increase in the number of available channels. Therefore, the deployment of these channel assignment strategies, in current cellular systems, seems well within the range of today's technology. On comparing the performance of these algorithms with that of fixed channel assignment, it is found that, for the interesting range of blocking probability, viz. 2 – 4%, an increase in carried traffic of about 60 – 80% can be obtained. This can be seen from Fig. 2 for the uniform traffic case, where the performance of the fixed channel assignment (8 channels per cell), is also plotted. A similar increase is obtained in the non-uniform traffic case also, where the best fixed channel assignment, obtained with the help of the algorithms described in [4], was used.

Furthermore, blocked calls were assumed to be cleared from the system. The carried traffic could be increased further by queuing the blocked calls.

In all of the above, the computation is centralized. It remains to be investigated as to how this may be distributed among the individual cell-sites, or the mobiles, or both.

References

[1] Everitt, D., and Manfield, D., "Performance Analysis of Cellular Mobile Communication Systems with Dynamic Channel Assignment", *IEEE J. Sel. Areas Commn.*, vol. 7, pp. 1172–1180, Oct. 1989.

[2] Gamst, A., "Some Lower Bounds for a Class of Frequency Assignment Problems", *IEEE Trans. Veh. Tech.*, vol. VT-35, pp. 8–14, Feb. 1986.

[3] MacDonald, V.H., "The Cellular Concept", *The Bell System Tech. J.*, vol. 58, pp. 15–41, Jan. 1979.

[4] Sivarajan, K. N., McEliece, R. J., and Ketchum, J. W., "Channel Assignment in Cellular Radio", *39th IEEE Veh. Tech. Conf.*, May 1989.

Appendix

Consider the following cellular system consisting of two cells with the separation matrix $\mathbf{C} = (c_{ij})$ given by :

$$\mathbf{C} = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix} .$$

Let $N_f = 3$. The allowed states of the system $\mathcal{N}$ are (0,0), (1,0), (0,1), (2,0), (0,2), (2,1) and (1,2).

It can be shown that the blocking probability for this cellular system is given by

$$\Pr\{ \text{ Incoming call blocked } \} = \frac{\rho^2(1-3p_1p_2) + p_1p_2\rho^3}{2+2\rho+\rho^2+p_1p_2\rho^3} .$$

These two cells constitute a clique with $N_{\text{clique}} = 3$. The lower bound on the blocking probability is given by the ErlangB formula with 3 servers viz.

$$\text{Lower Bound on Blocking Probability} = \frac{\rho^3/3!}{1+\rho+\rho^2/2!+\rho^3/3!} .$$

Let $p = p_1$. Then, $p_1p_2 = p(1-p)$, is a function only of p, and p serves as a measure of the degree of non-uniformity of the traffic. The above expressions are plotted in Fig. 4 as a function of the traffic for various degrees of non-uniformity, i.e., for various values of p. (The lower bound is independent of p.) The weakening of the lower bound, as the non-uniformity in the traffic increases, is evident from this plot.

Table 1 : Non-uniform Spatial Distribution of Traffic considered

i	p_i
1	0.0166
2	0.0520
3	0.0166
4	0.0166
5	0.0166
6	0.0312
7	0.0374
8	0.1081
9	0.1601
10	0.0582
11	0.0270
12	0.0312
13	0.0644
14	0.0312
15	0.0748
16	0.1185
17	0.0582
18	0.0166
19	0.0208
20	0.0270
21	0.0166

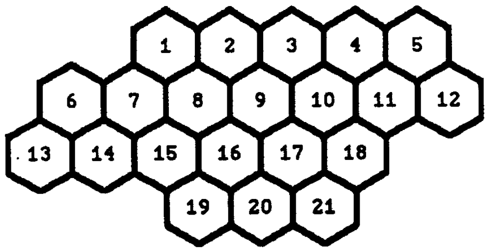

Fig. 1 : The 21-cell system
(The cell number is indicated within each cell.)

Table 2 : 1-cliques in the 21-cell system

Clique 1 = 9 2 8 16 17 20 3 10 15 19 18 21
Clique 2 = 9 2 8 16 17 20 3 10 15 19 1 7
Clique 3 = 9 2 8 16 17 20 3 10 11 4 18 21
Clique 4 = 9 2 8 16 17 20 3 10 4 1
Clique 5 = 9 2 8 16 17 20 14 1 15 19 7
Clique 6 = 9 2 8 16 6 1 14 15 7 19
Clique 7 = 9 2 5 4 3 10 11 17 18 21
Clique 8 = 9 12 4 5 10 11 3 17 18 21
Clique 9 = 13 6 7 8 1 14 15 16 19

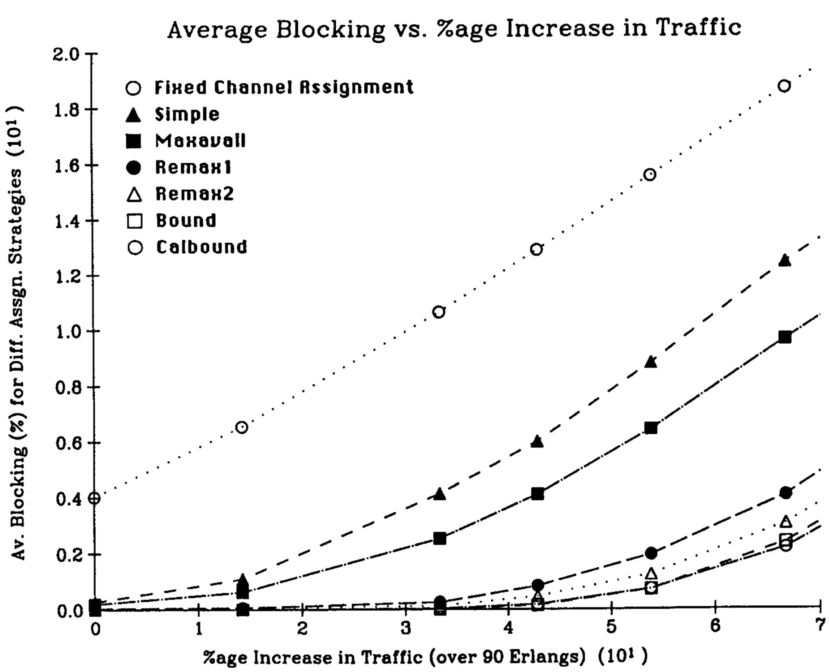

Fig. 2 : Uniform Traffic case

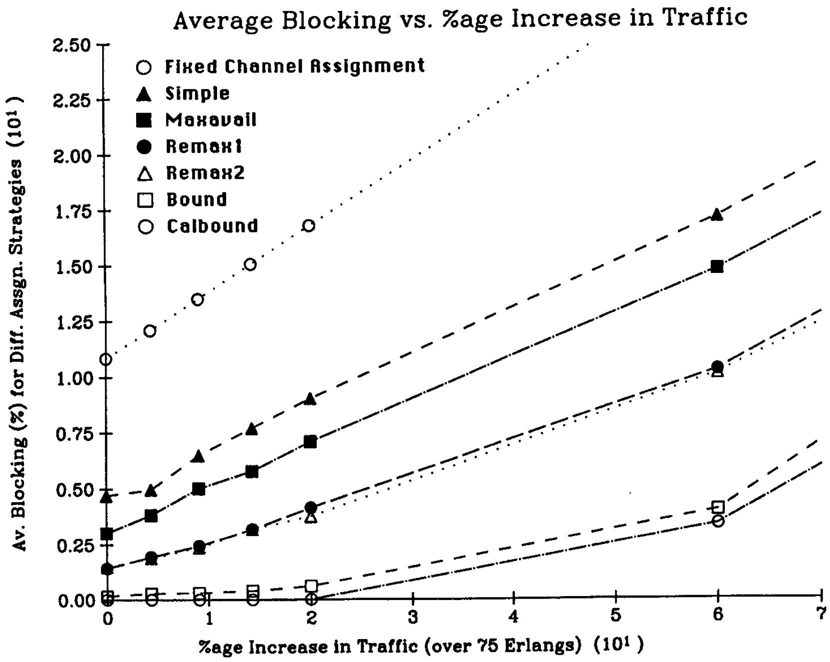

Fig. 3 : Non-uniform Traffic case

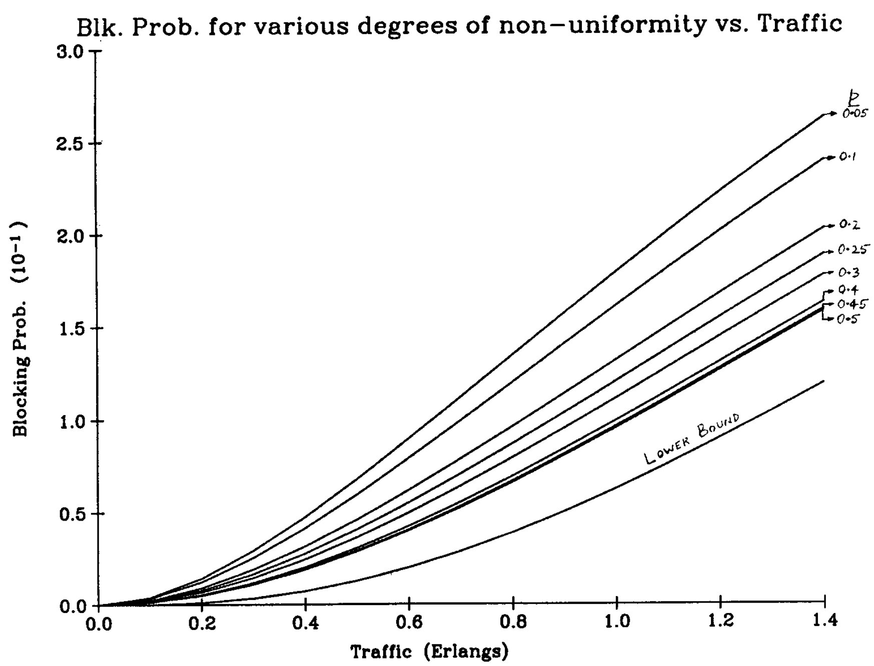

Fig. 4 : Blocking Probability for Various Degrees of Inhomogeneity